Journal of Soft Computing in Civil Engineering 8-2 (2024) 119-142

Contents lists available at SCCE

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

A Novel Center of Mass Optimization (CMO) Algorithm for Truss Design Problems

Hesam Varaee [1*]

1. Assistant Professor, Department of Engineering, Ale Taha Institute of Higher Education, Tehran, Iran
Corresponding author: *varaee.hesam@aletaha.ac.ir*

https://doi.org/10.22115/SCCE.2023.398542.1649

ARTICLE INFO

Article history:
Received: 22 May 2023
Revised: 15 August 2023
Accepted: 20 November 2023

Keywords:
Center of mass;
Population-based algorithms;
Mathematical and engineering optimization problems;
Wilcoxon rank-sum test;
ANOVA test.

ABSTRACT

This paper introduces a novel optimization algorithm rooted in the mass center equations of particle systems. The proposed Center of Mass Optimization (CMO) algorithm is distinguished by its easy implementation, parameter independence, and rapid, accurate solutions. In the proposed CMO, a random walk operator is introduced to enhance the exploitation capability of the CMO and help the search agents jump out of the local optimal. Mutation and elitism selection operators are also used to boost the overall performance of the proposed algorithm. Some mathematical benchmark optimization problems and two engineering truss optimization examples are investigated to evaluate the performance of the proposed method. The results are compared with those of well-known optimization algorithms such as DE, ABC, CBO, PSO, EO, LHHA, and SMA. The results of Wilcoxon rank-sum and ANOVA tests indicate that the performance of the proposed algorithm is robust and reliable for a wide range of complex mathematical and engineering optimization problems.

How to cite this article: Varaee A. A novel center of mass optimization (CMO) algorithm for truss design problems. J Soft Comput Civ Eng 2024;8(2):119–142. https://doi.org/10.22115/scce.2023.398542.1649

2588-2872/ © 2024 The Authors. Published by Pouyan Press.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Reducing material consumption and construction costs while meeting standard requirements has always been a critical challenge in civil and structural engineering [1–4]. In recent years, the development of optimization algorithms and their increasing application in civil engineering have made structural optimization popular for solving complex optimization problems [5–9]. Numerous studies have shown the exceptional performance of modern metaheuristic algorithms in solving highly nonlinear structural optimization problems, effectively reducing the weight and cost of structures [10,11]. These studies utilized various groups of metaheuristic optimization algorithms, including evaluation-based, swarm-based, physics-based, and human-based algorithms [12–14]. These algorithms have been successfully employed to solve complex engineering optimization problems, such as optimizing the cost of steel and concrete frames, determining the optimal weight of truss structures, performance-based design optimization (PBDO), reliability-based design optimization (RBDO), and life cycle cost (LCC) optimization of structures [15].

Evolution-based algorithms (EAs), which stimulate natural evolution such as migration, chemotaxis, and reproduction elimination and dispersal, have played a significant role in structural optimization. Among these, the genetic algorithm (GA) proposed by Holland is a widely used and well-known EA in the field of structural optimization [16,17]. The GA stands out for its ability to perform optimization in the search space without requiring explicit derivation, a characteristic often present in mathematical optimization approaches. The GA emulates the concept of the survival of the fittest in nature to evolve a population. Over the years, the GA has been improved, and numerous variants have been developed, enabling it to avoid local optima and provide efficient solutions. Other proposed evolutionary algorithms, such as harmony search (HS), differential evolution (DE), and stochastic fractal search (SFS), have also been successfully applied to solve engineering problems. Physics-based algorithms mimic physical laws in the universe and have found wide application in engineering problems. One such algorithm is simulated annealing (SA), which is based on thermodynamic principles of thermodynamics [18–20]. SA imitates the annealing process in materials, were heating and subsequent cooling leads to crystallization and energy minimization. Several novel physics-inspired techniques have been developed, including water wave optimization (WWO), vortex search algorithm (VSA), ideal gas molecular movement (IGMM) algorithm, and gravitational search algorithm (GSA). Swarm-based algorithms simulate social behaviors in species, such as the division of labor and self-organization. Notable instances encompass ant colony optimization (ACO) and particle swarm optimization (PSO). PSO mimics the flocking patterns of birds to iteratively update each agent within a population, drawing upon both the top-performing global agent and the most successful individual agent [21]. Meanwhile, ACO takes inspiration from the foraging conduct of ant colonies, where ants discover the most efficient route to food sources by responding to pheromone concentrations.

Other swarm-inspired algorithms include African buffalo optimization (ABO), joint operations algorithm (JOA), coyote optimization algorithm (COA), emperor penguin colony (EPC), gray wolf optimization (GWO), and shark smell optimization (SSO). In recent years, new metaheuristic algorithms, inspired by social behaviors and human ideologies, have emerged. These include political optimizer (PO), heap-based optimizer (HBO), social engineering optimizer (SEO), artificial human optimization (AHO), and parliamentary optimization algorithm (POA), to name a few.

This study presents a new algorithm that uses a novel approach. The proposed algorithm is inspired by the physical equation of the center of mass of a particle system. In addition, the algorithm incorporates a random walk operator, mutation operator, and elitism selection to enhance the local search and overall performance. A notable feature of the proposed algorithm is its parameter independence, which alleviates the challenge and time-consuming nature of parameter adjustment while significantly impacting the performance. To assess the performance, twenty-three mathematical benchmark functions were evaluated [22,23], and each problem was independently solved 20 times to reduce the uncertainty of the results. The results are compared with those of well-known and powerful metaheuristic algorithms, such as DE, ABC, CBO, PSO, EO, LHHA and SMA. The Wilcoxon rank-sum and ANOVA tests indicate that the proposed CMO algorithm outperforms other algorithms [10,12]. Two examples of engineering truss problems, one in two dimensions and the other in three dimensions, were also examined to evaluate the performance of the algorithm in complex engineering problems. Statistical results from 20 independent runs for each truss problem demonstrated the satisfactory performance of the CMO algorithm. Solving highly nonlinear engineering problems often entails high computational costs and requires a substantial number of analyses. Compared with other algorithms, the CMO algorithm exhibits a slight standard deviation and requires fewer analyses to achieve the optimal solution, indicating its suitability for complex and highly nonlinear engineering problems.

The remainder of this paper is organized as follows. Section 2 presents the procedure of the proposed CMO algorithm. Sections 3 and 4 evaluate mathematical and engineering truss problems, respectively. Finally, Section 5 concludes the paper and highlights the key findings.

2. New proposed CMO algorithm

2.1. Initialization

In this study, a group of particles with different masses, each representing a potential solution in the problem domain, was produced randomly with a uniform distribution within the allowable range of the problem variables. To generate the initial population, Eq. 1 is used.

$$\bar{X} = unifrnd\ (X_{min}.X_{max}.d) \tag{1}$$

In the above equation, $\bar{X}$ represents the set of initial particles, X_{min} and X_{max} denote the upper and lower bounds of the variables, respectively, and d represents the dimension of the problem, or in other words, the number of design variables. The $unifrnd$ function randomly generates particles with a uniform distribution.

2.2. Mass assignment

After the particles were evaluated, a mass corresponding to the fitness of each particle was assigned. Therefore, the fittest particles will have a larger mass and vice versa. The mass was assigned to the particles using Eq. 2.

$$m_i = \frac{1}{abs(C_i - best)/abs(worst - best)} \quad (2)$$

In Eq. 2, m_i represents the mass of the i-th particle, C_i indicates the fitness of the i-th particle, and $best$ and $worst$ represent the best and worst particles found in the population, respectively.

2.3. Center of mass calculation

The center-of-mass relations of a set of particles are used as the main operator of the CMO algorithm. The center of mass of a system of particles is the point at which the entire mass of the system is concentrated when it moves, and all external forces affect that point. Fig. 1 illustrates a system of n particles with different masses and positions.

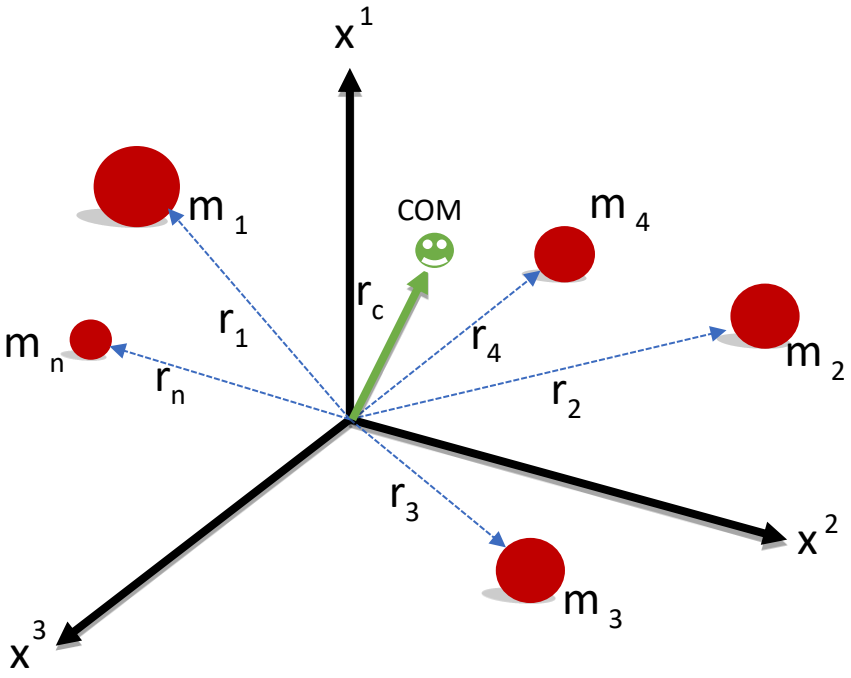


Fig. 1. Center of mass of a particle system.

According to Fig. 1, the coordinates of the center of mass of a three-dimensional system consisting of n particles are obtained from Eq. 3:

$$r_{com} = \frac{1}{M}\sum_{i=1}^{n} m_i r_i \tag{3}$$

In the above equation, r_{com} represents the position vector of the center of mass of the particle system. m_i and r_i represent the mass and position vectors, respectively, of each particle. M is the sum of the masses of all the particles, that is, the mass of the entire system, and is obtained from Eq. 4:

$$M = \sum_{i=1}^{n} m_i \tag{4}$$

Similar equations can be written based on the coordinates of each particle in the d-Dimention Cartesian system as follows:

$$\begin{aligned} x_{com}^{1} &= \frac{1}{M}\sum_{i=1}^{n} m_i x_i^1 \\ x_{com}^{2} &= \frac{1}{M}\sum_{i=1}^{n} m_i x_i^2 \\ x_{com}^{3} &= \frac{1}{M}\sum_{i=1}^{n} m_i x_i^3 \\ &\dots \\ x_{com}^{d} &= \frac{1}{M}\sum_{i=1}^{n} m_i x_i^d \end{aligned} \tag{5}$$

In Eq. 5, superscripts 1 to d represent the first d-th dimension of the problem.

The main loop of the proposed CMO algorithm includes global and local searches, and is enhanced by mutation and elitism operators. To search the entire problem space, in the first step of the algorithm's main loop, the center of mass of each particle and the best particle found thus far are calculated using Eq. 6.

$$X_{ci} = \frac{rand \times X_i \times m_i + rand \times X_{best} \times m_{best}}{rand \times (m_i + m_{best})} \tag{6}$$

where X_{ci} is the center of mass between the current particle and the best particle of the population. $rand$ represents a random number with a normal distribution between zero and one.

In the next step, by comparing the fitness of X_{ci} and the fitness of X_i , the best particle is selected and replaced instead of the current particle by comparing the fitness of Xci and Xi. It should be noted that X_{best} has the largest mass compared with the other particles, and X_{ci} is always

inclined towards the best particle of the system. Therefore, because of the higher chance of finding better particles in the vicinity of the best particle, repeating the above process can cause convergence of the system to the optimal region, and this process can be called a global search.

2.4. Random-walk operator

Then, a random-walk operator in the form of Eq. Seven was used to perform a local search. The random walk operator generates a new particle adjacent to the current particle.

$$X_{li} = X_i + \delta_i \tag{7}$$

where X_{li} is the newly generated particle and δ_i is a partial random value with a uniform distribution within the range of -1 to 1. The random-walk operator can be considered a simulator for various natural phenomena. These include the Brownian motion of molecules in a liquid or gas, the search for food-seeking behavior of animals, the volatility of stock prices, and the investor's financial condition. By comparing the fitness of the newly generated particle and the fitness of the current particle, the best one was selected to continue the optimization process.

2.5. Mutation operator

A mutation operator can be helpful in avoiding local optima. Therefore, some particles (with a probability of rand > 0.8) underwent mutations using Eq. 8:

$$X_i = X_i + \Delta \tag{8}$$

where Δ is a random value with a uniform distribution within the range -2 to 2.

Finally, the elitism operator preserves the best particle ever found and carries it to the next generation. Fig. 2 shows the flowchart of the proposed CMO algorithm.

3. Numerical and examples

3.1. mathematical benchmark functions

This study investigated 23 mathematical benchmark functions described in Tables 1-3 to evaluate the performance of the proposed algorithm. Mathematical functions are divided into three different groups: unimodal, multimodal, and multimodal functions with fixed dimensions. The unimodal functions (F_1 to F_6) have a single global optimum and are used to evaluate the global search ability, exploration, and convergence speed of the algorithms. The multimodal functions (F_7 to F_{14}) have multiple local optima, with the best one being the global optimum. Algorithms that can effectively balance exploration and exploitation can successfully handle these functions. Functions F_{15} to F_{23} are multimodal functions with fixed dimensions. These functions are typically simple examples with different variable ranges and can be used to assess the accuracy and robustness of algorithms [24,25].

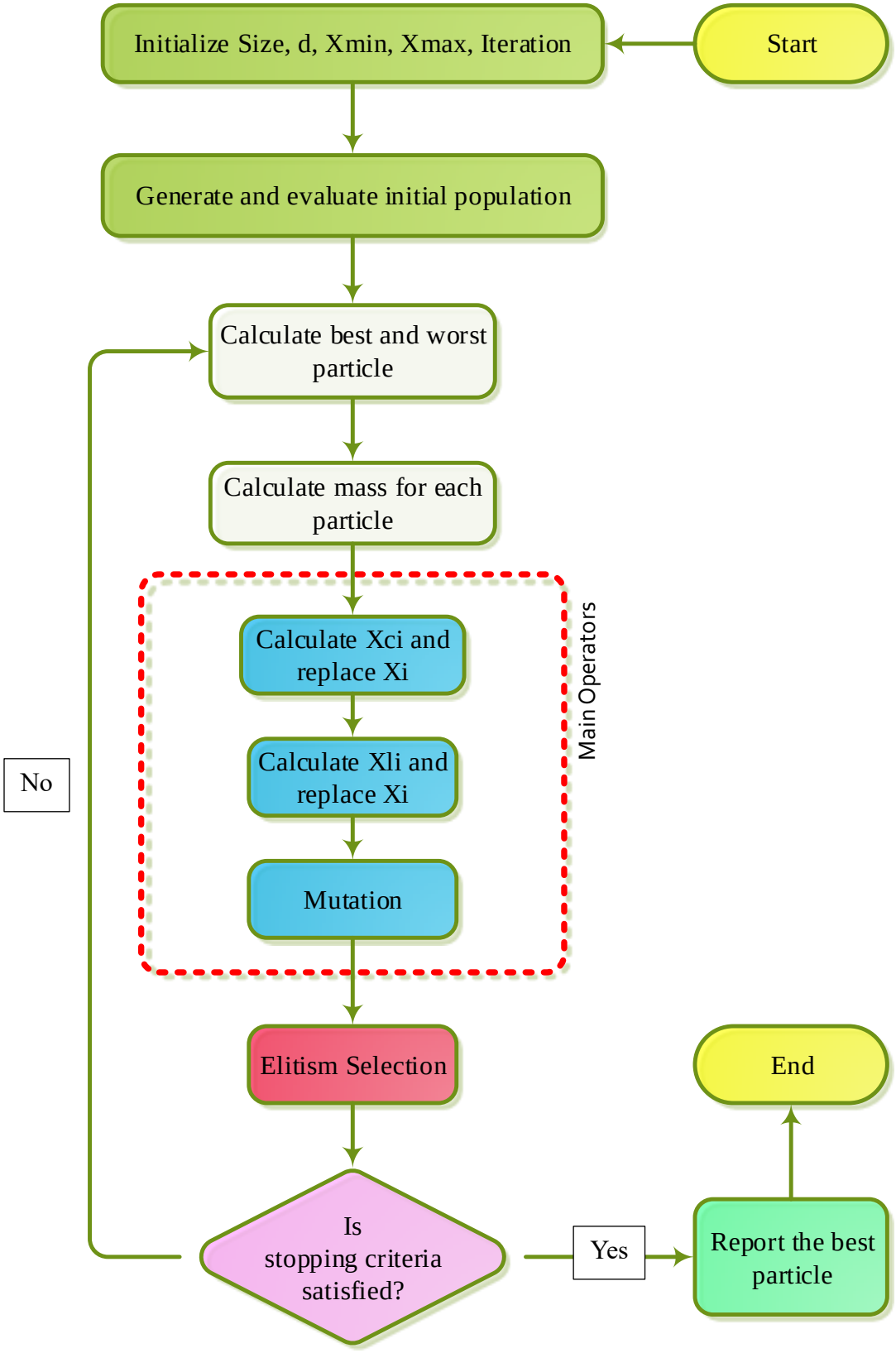


Fig. 2. The flowchart of the proposed CMO.

Table 1
Unimodal benchmark functions.

Type	Test Function	n	S	f_{min}
Unimodal	$F_1(X) = \sum_{i=1}^{n} x_i^2$	30	$[-100.100]^n$	0
Unimodal	$F_2(X) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10.10]^n$	0
Unimodal	$F_3(X) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$[-100.100]^n$	0
Unimodal	$F_4(X) = \max_i \{ \lvert x_i \rvert . 1 \le i \le n \}$	30	$[-100.100]^n$	0
Unimodal	$F_5(X) = \sum_{i=1}^{n-1} [100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2]$	30	$[-30.30]^n$	0
Unimodal	$F_6(X) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	$[-100.100]^n$	0

Table 2
Multimodal benchmark functions.

Type	Test Function	n	S	f_{min}
Multimodal	$F_7(X) = \sum_{i=1}^{n} i{x_i}^4 + \text{random}[0.1)$	30	$[-1.28.1.28]^n$	0
Multimodal	$F_8(X) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500.500]^n$	-418.98×30
Multimodal	$F_9(X) = \sum_{i=1}^{n} [{x_i}^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12.5.12]^n$	0
Multimodal	$F_{10}(X) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	30	$[-32.32]^n$	0
Multimodal	$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600.600]^n$	0
Multimodal	$F_{12}(X) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i. 10.100.4)$, $y_i = 1 + \frac{x_i + 1}{4}$, $u(x_i.a.k.m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50.50]^n$	0
Multimodal	$F_{13}(X) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\}$	30	$[-50.50]^n$	0
Multimodal	$F_{14}(X) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	$[-65,65]$	1

Table 3
Multimodal benchmark functions with fixed dimensions.

Type	Test Function	n	S	f_{min}
Multimodal-Fixed dimensions	$F_{15}(X)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	4	$[-5,5]$	0.00030
Multimodal-Fixed dimensions	$F_{16}(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$[-5,5]$	-1.0316
Multimodal-Fixed dimensions	$F_{17}(X)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10\left(1-\frac{1}{8\pi}\right)cosx_1+10$	2	$[-5,5]$	0.398
Multimodal-Fixed dimensions	$F_{18}(X)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	$[-2,2]$	3
Multimodal-Fixed dimensions	$F_{19}(X)=-\sum_{i=1}^{4}c_i\,exp(-\sum_{j=1}^{3}a_{ij(}\,x_j-p_{ij)}{}^2)$	3	[1,3]	-3.86
Multimodal-Fixed dimensions	$F_{20}(X)=-\sum_{i=1}^{4}c_i\,exp(-\sum_{j=1}^{6}a_{ij}\,(x_j-p_{ij})^2)$	6	[0,1]	-3.32
Multimodal-Fixed dimensions	$F_{21}(X)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
Multimodal-Fixed dimensions	$F_{22}(X)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
Multimodal-Fixed dimensions	$F_{23}(X)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.5363

To compare the results of the CMO algorithm with those of other well-known algorithms, each problem was evaluated 20 times independently using the DE, ABC, CBO, PSO, EO, LHHA and SMA algorithms in addition to the CMO. To ensure fair comparisons and identical conditions among all algorithms, the number of particles for all algorithms was set to 50, and the maximum number of optimization cycles was set to 100. A distinguishing feature of the CMO algorithm is its lack of dependence on initial parameter adjustments. The parameter settings for the other optimization algorithms, as listed in Table 4, were established based on the values proposed in the original references for each algorithm.

Table 4
Parameter settings of optimization algorithms.

Algorithm	Parameters
DE [26]	Beta-min=0.1, Beta-max=0.2, PCR=0.2
ABC [27]	FS=NB/2, ne=NB/2, no=NB/2, Scout=1, Limit=ne×D
CBO [28]	COR=1-(iter/maxIt)
PSO [29]	V_{max}=5, W_{max}, W_{min}=0.4-0.9, C1,C2=2
EO [30]	V=1, α_1=2, α_2=1, GP=0.5
LHHA [31]	E_0=2×rand()-1, E_1=2× (1-(iter/maxIt))
SMA [32]	z=0.03, a = $\tan^{-1}$(-(iter/ maxIt)+1, b = 1-(iter/ maxIt)

The comparison results for each algorithm are presented in Tables 5–7. The results demonstrate that the CMO algorithm outperforms the other algorithms in 16 of the 23 functions. In addition, the CMO algorithm exhibits a small standard deviation, highlighting its high robustness and stability. This characteristic is particularly crucial for solving complex and large-scale problems that cannot be solved repeatedly.

Table 5
Statistical results for the unimodal benchmark functions.

Func.		CMO	DE	ABC	CBO	PSO	EO	LHHA	SMA
F_1	Best	**5.07E-151**	1.21E+03	2.87E+00	3.19E+02	2.15E+03	7.74E-08	1.17E-49	2.19E-81
	Average	**2.30E-83**	1.68E+03	6.28E+02	9.83E+02	4.92E+03	9.95E-07	5.61E-41	1.52E-58
	Worst	**4.13E-82**	2.60E+03	3.15E+03	1.77E+03	1.22E+04	3.04E-06	9.28E-40	3.05E-57
	Std.	**9.24E-83**	3.52E+02	8.78E+02	4.03E+02	3.00E+03	8.71E-07	2.09E-40	6.82E-58
F_2	Best	**1.69E-80**	1.27E+01	5.47E-01	6.34E+00	3.55E+01	4.74E-05	3.43E-27	1.81E-45
	Average	**1.04E-48**	1.50E+01	9.44E-01	9.83E+00	7.38E+01	9.64E-05	1.46E-21	3.52E-17
	Worst	**1.95E-47**	1.84E+01	2.47E+00	2.28E+01	1.17E+02	1.70E-04	1.25E-20	7.04E-16
	Std.	**4.36E-48**	1.49E+00	4.30E-01	3.93E+00	2.11E+01	3.20E-05	2.97E-21	1.57E-16
F_3	Best	**1.14E-126**	3.60E+04	2.12E+04	7.75E+03	2.98E+04	2.53E-01	4.21E-40	5.05E-96
	Average	**1.00E-46**	4.72E+04	3.31E+04	1.51E+04	4.39E+04	5.62E+00	1.80E-27	5.38E-60
	Worst	**2.01E-45**	6.41E+04	4.73E+04	2.33E+04	6.84E+04	2.04E+01	1.81E-26	1.07E-58
	Std.	**4.49E-46**	7.37E+03	5.35E+03	3.87E+03	9.75E+03	6.22E+00	4.72E-27	2.38E-59
F_4	Best	**1.16E-64**	4.91E+01	5.76E+01	3.14E+01	5.13E+01	2.01E-02	2.69E-25	2.12E-45
	Average	**5.29E-40**	5.79E+01	7.20E+01	4.90E+01	6.35E+01	7.50E-02	2.37E-21	1.94E-31
	Worst	**4.00E-39**	6.54E+01	7.84E+01	6.49E+01	7.68E+01	2.01E-01	4.23E-20	3.73E-30
	Std.	**1.30E-39**	3.81E+00	5.30E+00	1.07E+01	6.25E+00	4.59E-02	9.41E-21	8.33E-31
F_5	Best	1.17E-01	3.58E+05	1.94E+02	9.33E+04	9.79E+05	2.70E+01	**3.84E-06**	4.63E-06
	Average	1.51E+01	6.54E+05	8.50E+02	5.44E+05	4.60E+06	2.75E+01	3.22E+00	**5.24E-04**
	Worst	2.89E+01	1.16E+06	3.00E+03	2.04E+06	1.08E+07	2.85E+01	2.87E+01	**2.75E-03**
	Std.	1.23E+01	2.10E+05	9.09E+02	4.20E+05	2.75E+06	3.91E-01	8.78E+00	**6.67E-04**
F_6	Best	3.95E-02	9.39E+02	3.10E+00	5.33E+02	1.98E+03	3.12E-02	2.27E-04	**9.64E-08**
	Average	4.93E-01	1.64E+03	1.19E+03	9.74E+02	6.53E+03	3.37E-01	3.02E-03	**9.87E-06**
	Worst	1.33E+00	2.22E+03	4.25E+03	1.63E+03	1.50E+04	1.07E+00	2.13E-02	**1.19E-04**
	Std.	3.79E-01	3.58E+02	1.29E+03	2.92E+02	4.53E+03	2.68E-01	4.75E-03	**2.63E-05**

Table 6
Statistical results for the multimodal benchmark functions.

Func.		CMO	DE	ABC	CBO	PSO	EO	LHHA	SMA
F_7	Best	**7.27E-05**	5.16E-01	5.08E-01	2.71E-01	7.73E-01	1.75E-03	1.43E-05	5.98E-03
	Average	**6.66E-04**	7.50E-01	1.07E+00	6.73E-01	5.09E+00	4.70E-03	5.80E-04	3.64E-02
	Worst	**2.07E-03**	9.98E-01	2.02E+00	2.17E+00	2.59E+01	8.72E-03	1.70E-03	1.10E-01
	Std.	**5.39E-04**	1.40E-01	3.71E-01	4.64E-01	5.58E+00	2.06E-03	5.88E-04	2.45E-02
F_8	Best	**-12569.48**	-6851.74	-8593.80	-5417.67	-9293.97	-8959.75	-12569.49	-9913.27
	Average	**-12562.00**	-5977.60	-8098.83	-4598.40	-8226.00	-7821.54	-12554.33	-5207.09
	Worst	**-12544.81**	-5426.06	-7644.57	-3782.71	-6957.59	-6380.28	-12330.16	-3131.94
	Std.	**6.92E+00**	3.64E+02	3.10E+02	5.60E+02	5.16E+02	6.57E+02	5.30E+01	2.21E+03
F_9	Best	**0.00E+00**	1.66E+02	7.54E+01	7.70E+01	1.65E+02	1.87E-06	0.00E+00	0.00E+00
	Average	**0.00E+00**	1.83E+02	9.55E+01	1.43E+02	2.58E+02	7.73E-01	0.00E+00	0.00E+00
	Worst	**0.00E+00**	1.99E+02	1.14E+02	2.25E+02	3.09E+02	6.81E+00	0.00E+00	0.00E+00
	Std.	**0.00E+00**	9.33E+00	1.32E+01	4.33E+01	4.12E+01	1.60E+00	0.00E+00	0.00E+00
F_{10}	Best	**8.88E-16**	8.84E+00	1.13E+01	5.39E+00	9.00E+00	7.75E-05	8.88E-16	8.88E-16
	Average	**8.88E-16**	9.87E+00	1.39E+01	7.67E+00	1.32E+01	1.81E-04	8.88E-16	1.07E-15
	Worst	**8.88E-16**	1.11E+01	1.59E+01	9.47E+00	1.82E+01	4.42E-04	8.88E-16	4.44E-15
	Std.	**0.00E+00**	5.79E-01	1.07E+00	9.58E-01	2.83E+00	8.68E-05	0.00E+00	7.94E-16
F_{11}	Best	**0.00E+00**	8.90E+00	9.87E-01	4.02E+00	1.67E+01	5.99E-07	0.00E+00	0.00E+00
	Average	**0.00E+00**	1.58E+01	7.13E+00	9.03E+00	4.13E+01	8.60E-04	0.00E+00	0.00E+00
	Worst	**0.00E+00**	2.21E+01	1.89E+01	1.38E+01	1.17E+02	1.72E-02	0.00E+00	0.00E+00
	Std.	**0.00E+00**	3.17E+00	6.39E+00	2.67E+00	2.35E+01	3.84E-03	0.00E+00	0.00E+00
F_{12}	Best	1.20E-04	2.19E+02	2.59E-03	3.62E+01	5.94E+04	8.95E-04	2.65E-05	**4.36E-08**
	Average	2.71E-02	2.16E+04	7.36E-02	1.01E+05	2.63E+06	1.14E-02	4.38E-04	**1.55E-06**
	Worst	1.08E-01	1.88E+05	2.13E-01	6.81E+05	1.02E+07	2.62E-02	3.04E-03	**1.12E-05**
	Std.	3.50E-02	4.10E+04	6.56E-02	1.66E+05	2.73E+06	8.40E-03	6.91E-04	**2.45E-06**
F_{13}	Best	2.47E-05	6.96E+04	3.30E-02	4.77E+04	5.05E+05	9.72E-02	5.11E-06	**2.87E-08**
	Average	3.97E-01	5.93E+05	3.23E-01	9.46E+05	6.84E+06	3.73E-01	1.04E-02	**1.70E-05**
	Worst	2.81E+00	1.96E+06	1.29E+00	3.51E+06	1.33E+07	6.25E-01	5.27E-02	**1.50E-04**
	Std.	7.82E-01	4.19E+05	2.91E-01	8.39E+05	4.38E+06	1.40E-01	1.64E-02	**3.34E-05**
F_{14}	Best	**9.98E-01**	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Average	**9.98E-01**	9.98E-01	9.98E-01	3.80E+00	9.98E-01	9.98E-01	1.10E+00	2.78E+00
	Worst	**9.98E-01**	9.98E-01	9.98E-01	8.88E+00	9.98E-01	9.98E-01	1.99E+00	9.80E+00
	Std.	**5.59E-05**	7.20E-17	6.91E-13	2.31E+00	6.99E-11	2.60E-16	3.06E-01	2.31E+00

Table 7
Statistical results for the multimodal with fixed dimension benchmark functions.

Func.		CMO	DE	ABC	CBO	PSO	EO	LHHA	SMA
F_{15}	Best	**4.20E-04**	7.98E-04	7.40E-04	7.27E-04	7.18E-04	3.37E-04	3.08E-04	3.20E-04
	Average	**7.78E-04**	1.01E-03	1.52E-03	2.05E-03	5.11E-03	6.67E-04	6.39E-04	4.49E-04
	Worst	**1.23E-03**	1.26E-03	2.19E-03	1.65E-02	2.04E-02	1.22E-03	1.43E-03	7.30E-04
	Std.	**2.51E-04**	1.37E-04	4.26E-04	3.49E-03	7.87E-03	2.27E-04	4.02E-04	1.31E-04
F_{16}	Best	**-1.0316**	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Average	**-1.0316**	-1.0316	-1.0316	-1.0311	-1.0316	-1.0316	-1.0316	-1.0311
	Worst	**-1.0316**	-1.0316	-1.0316	-1.0299	-1.0316	-1.0316	-1.0316	-1.0285
	Std.	**6.85E-08**	1.43E-09	1.53E-12	4.98E-04	1.07E-12	1.44E-16	6.24E-10	6.76E-04
F_{17}	Best	**0.3979**	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
	Average	**0.3979**	0.3979	0.3979	0.3996	0.3979	0.3979	0.3979	0.3981
	Worst	**0.3979**	0.3979	0.3979	0.4170	0.3979	0.3979	0.3979	0.4001
	Std.	**7.48E-07**	1.95E-06	4.72E-07	5.34E-03	4.73E-10	0.00E+00	4.01E-07	4.75E-04
F_{18}	Best	**3.0000**	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	Average	**3.0000**	3.0000	3.0035	3.0000	3.0000	3.0000	3.0000	3.0281
	Worst	**3.0000**	3.0000	3.0146	3.0000	3.0000	3.0000	3.0000	3.1154
	Std.	**2.97E-08**	3.67E-13	5.69E-03	1.80E-15	1.87E-11	2.68E-15	2.42E-07	3.60E-02
F_{19}	Best	**-3.8628**	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8625
	Average	**-3.8628**	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8601	-3.7968
	Worst	**-3.8627**	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8489	-3.6008
	Std.	**2.42E-05**	4.43E-15	1.04E-08	2.04E-15	2.09E-10	1.84E-15	3.98E-03	6.87E-02
F_{20}	Best	-3.3219	-3.3220	-3.3220	**-3.3220**	-3.3220	-3.3220	-3.2595	-3.1890
	Average	-3.2274	-3.3196	-3.3220	**-3.3220**	-3.2485	-3.2471	-3.1036	-2.9660
	Worst	-3.1244	-3.3048	-3.3218	**-3.3220**	-3.1299	-3.1376	-2.9227	-2.6053
	Std.	5.85E-02	4.03E-03	3.79E-05	**2.30E-07**	7.10E-02	6.44E-02	9.16E-02	1.50E-01
F_{21}	Best	-10.1527	-9.9878	**-10.1532**	-10.1532	-10.1532	-10.1532	-10.1456	-10.1384
	Average	-10.1441	-8.7810	**-10.1405**	-5.1580	-6.7809	-7.9998	-8.2304	-10.0436
	Worst	-10.0547	-5.6056	**-10.0827**	-2.6829	-2.6305	-2.6829	-5.0365	-9.8898
	Std.	2.24E-02	1.20E+00	**2.00E-02**	3.12E+0	3.82E+0	2.75E+00	2.40E+0	6.97E-02
F_{22}	Best	-10.4028	-10.4029	**-10.4029**	-10.4029	-10.4029	-10.4029	-10.4008	-10.3859
	Average	-9.8711	-9.6307	**-10.3243**	-10.0206	-7.5084	-9.4214	-7.3729	-10.3079
	Worst	-5.0873	-7.6669	-10.0223	-2.7659	-1.8376	-2.7659	-5.0766	**-10.1440**
	Std.	1.63E+00	8.32E-01	1.07E-01	1.71E+0	3.69E+0	2.43E+00	2.60E+0	**6.63E-02**
F_{23}	Best	**-10.5364**	-10.5363	-10.5352	-10.5364	-10.5364	-10.5364	-10.5362	-10.5296
	Average	**-10.5345**	-9.6576	-10.1501	-9.8779	-6.7265	-9.8146	-8.2751	-10.4536
	Worst	**-10.5293**	-5.4537	-5.0981	-3.1809	-2.4217	-2.8066	-5.1230	-10.1355
	Std.	**1.76E-03**	1.32E+00	1.20E+0	2.04E+0	3.65E+0	2.23E+00	2.64E+0	9.78E-02

Figs. 3-5 show the convergence curves of the problems. The significant difference between the CMO results and other algorithms results in both the accuracy and speed of convergence.

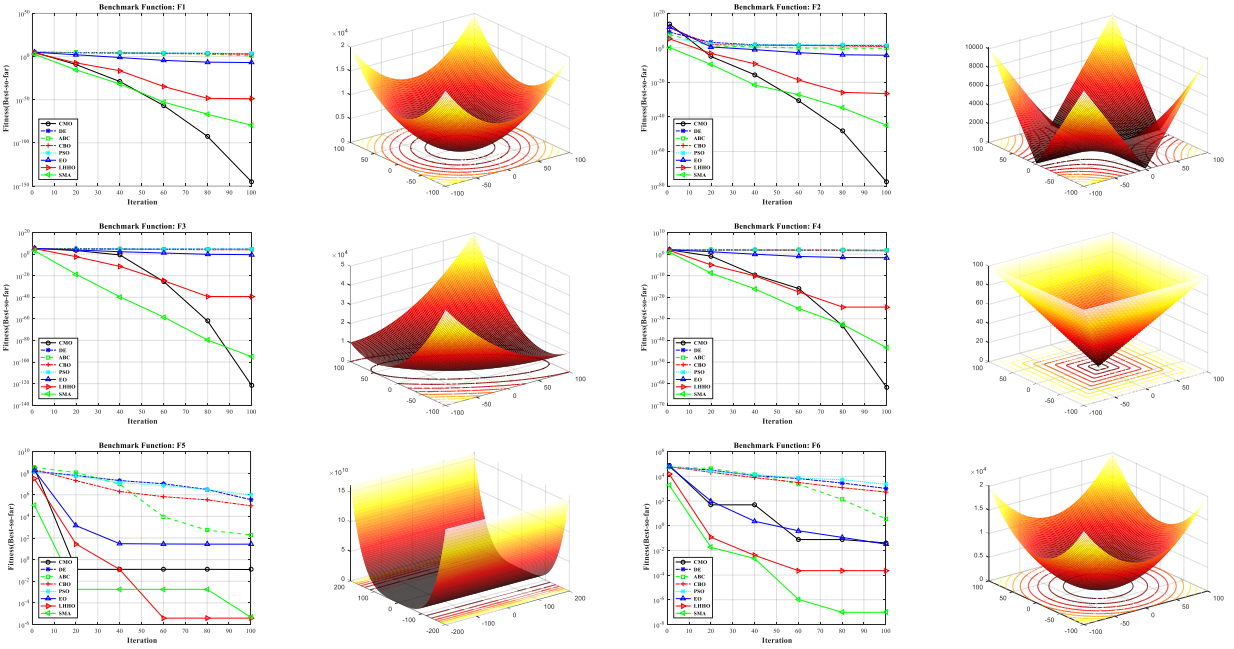


Fig. 3. The convergence curve of the unimodal benchmark problems.

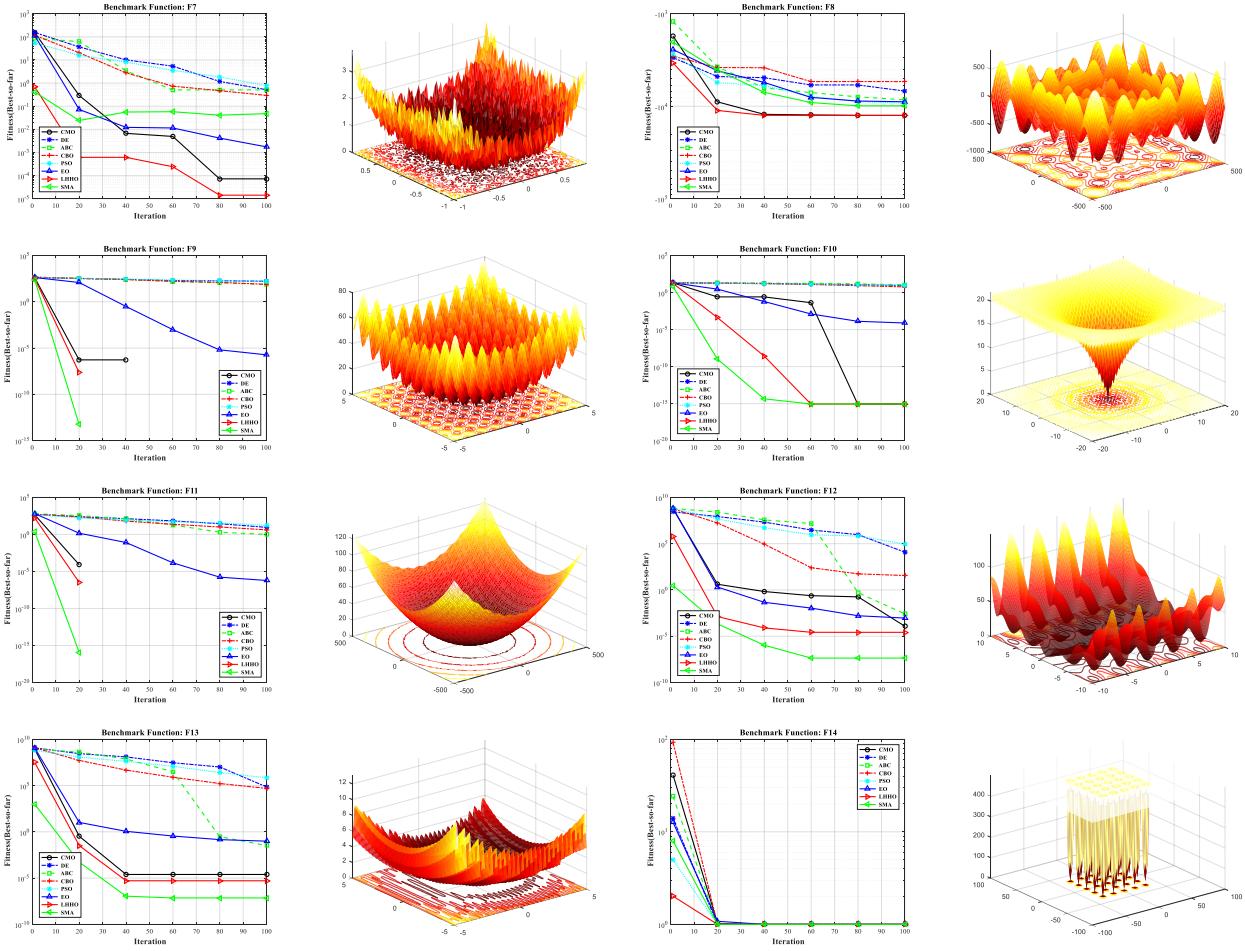


Fig. 4. The convergence curve of the multimodal benchmark problems.

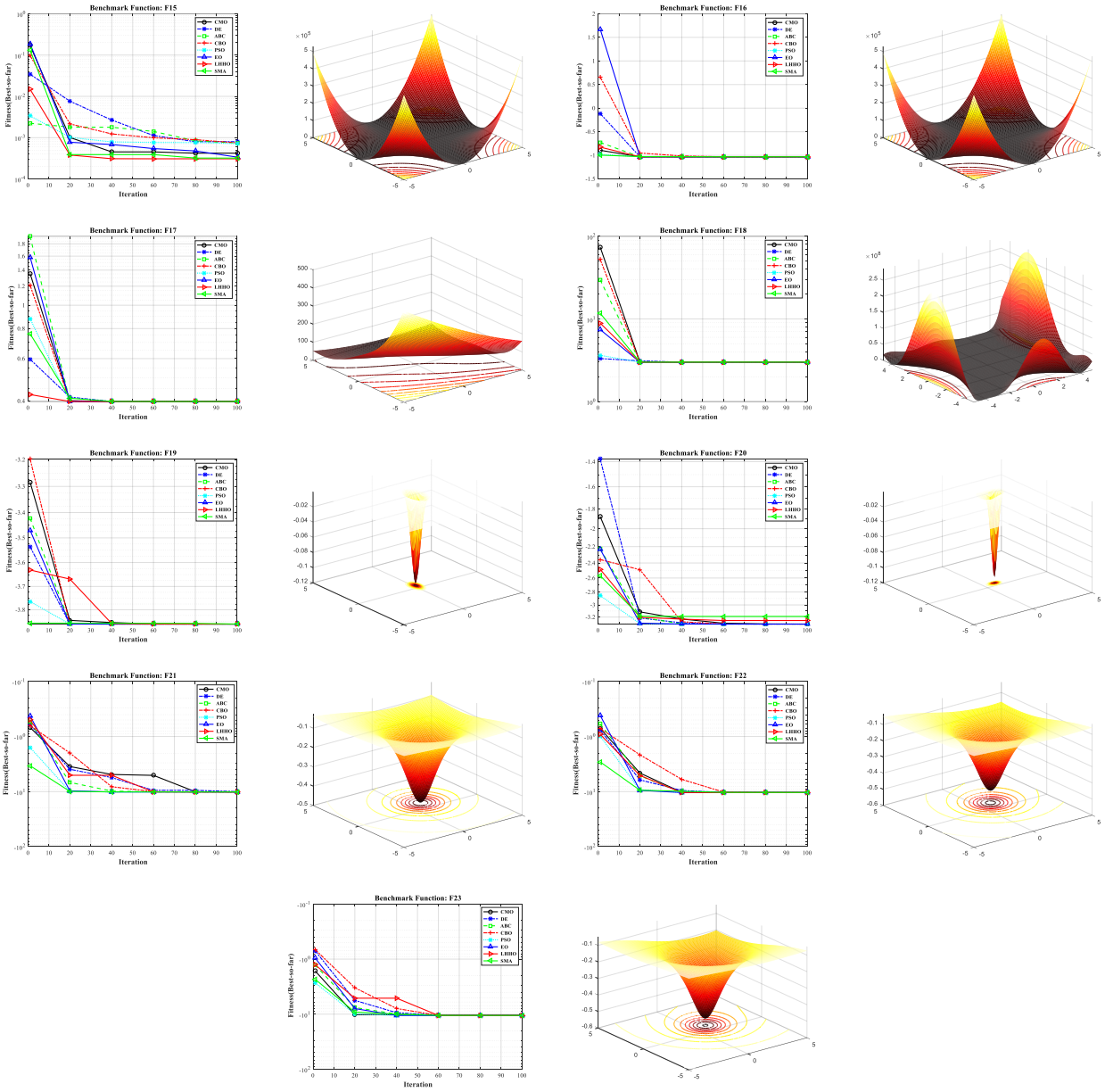


Fig. 5. The convergence curve of the multimodal benchmark problems with fixed dimensions.

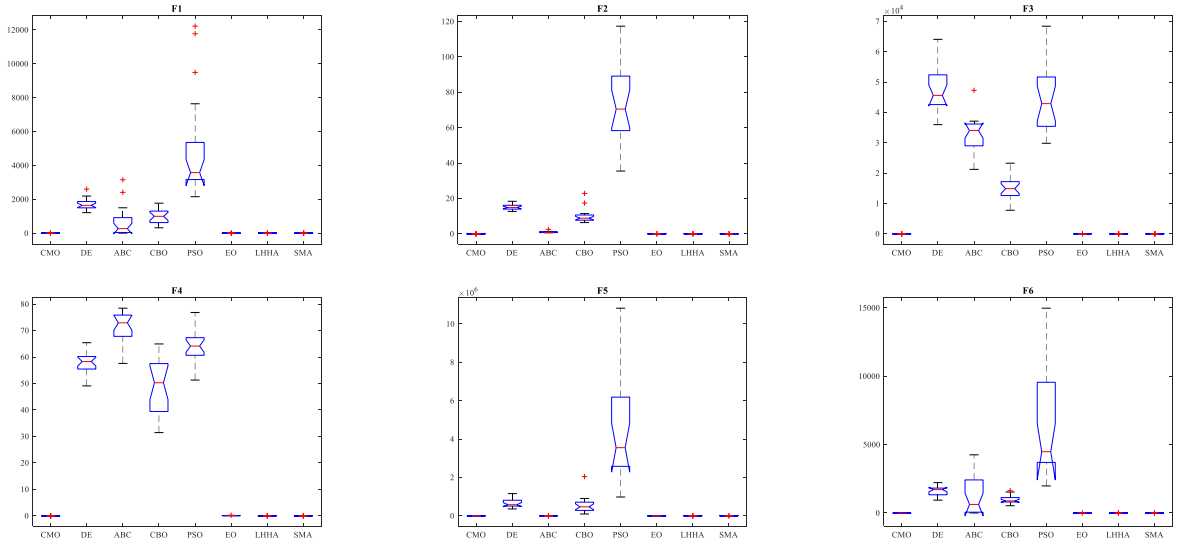
3.2. Wilcoxon rank-sum test

The Wilcoxon rank-sum test was employed to conduct a robust statistical comparison of the outcomes. In adherence to a significance level of 5%, Table 8 showcases the comprehensive results derived from this rigorous analysis. The discerned patterns solidify the position of the CMO algorithm as the standout performer among all algorithms under scrutiny.

Moreover, Fig. 6 provides a visual representation in the form of a boxplot, capturing the outcomes of the ANOVA test performed across the spectrum of the 23 mathematical benchmark functions. This graphical depiction enhances our understanding of the CMO algorithm's consistent efficacy across varied function landscapes.

Table 8
P-value of the Wilcoxon rank-sum test for mathematical benchmark functions.

Function	DE	ABC	CBO	PSO	EO	LHHA	SMA
F_1	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
F_2	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
F_3	6.80E-08	6.80E-08	6.80E-08	6.80E-08	8.35E-03	6.80E-08	6.80E-08
F_4	6.80E-08	6.80E-08	6.80E-08	6.80E-08	2.36E-06	6.80E-08	6.80E-08
F_5	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	2.30E-05	0.29
F_6	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	0.19
F_7	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	0.27	9.17E-08
F_8	6.80E-08	6.80E-08	6.61E-08	6.80E-08	6.80E-08	0.08	6.80E-08
F_9	8.01E-09	8.01E-09	8.01E-09	8.01E-09	8.01E-09	8.01E-09	8.01E-09
F_{10}	8.01E-09	8.01E-09	8.01E-09	8.01E-09	0.34	8.01E-09	8.01E-09
F_{11}	8.01E-09	8.01E-09	8.01E-09	8.01E-09	8.01E-09	8.01E-09	8.01E-09
F_{12}	6.80E-08	3.34E-03	6.80E-08	6.80E-08	6.80E-08	1.10E-05	0.82
F_{13}	6.80E-08	3.15E-02	6.80E-08	6.80E-08	1.06E-07	3.38E-04	3.97E-03
F_{14}	1.51E-08	1.57E-07	9.17E-08	6.58E-05	6.92E-07	2.07E-02	5.18E-08
F_{15}	1.63E-03	5.87E-06	2.22E-04	7.11E-03	8.60E-06	4.99E-02	0.10
F_{16}	1.69E-06	6.78E-08	1.38E-06	6.79E-08	6.80E-08	2.92E-07	3.96E-08
F_{17}	4.64E-05	8.60E-06	6.04E-06	7.89E-08	9.17E-08	7.38E-05	8.01E-09
F_{18}	6.79E-08	1.33E-02	5.35E-08	6.80E-08	6.80E-08	3.64E-03	6.71E-08
F_{19}	6.22E-08	6.80E-08	4.14E-08	6.80E-08	6.80E-08	2.30E-05	2.94E-08
F_{20}	1.78E-03	2.56E-07	6.80E-08	0.11	1.06E-07	9.75E-06	2.14E-03
F_{21}	6.80E-08	0.74	6.56E-03	0.80	3.94E-07	2.56E-07	0.29
F_{22}	5.09E-04	6.22E-04	1.60E-05	0.54	1.60E-05	3.50E-06	1.61E-04
F_{23}	1.20E-06	1.38E-06	1.60E-05	0.30	7.90E-08	1.05E-06	1.44E-04



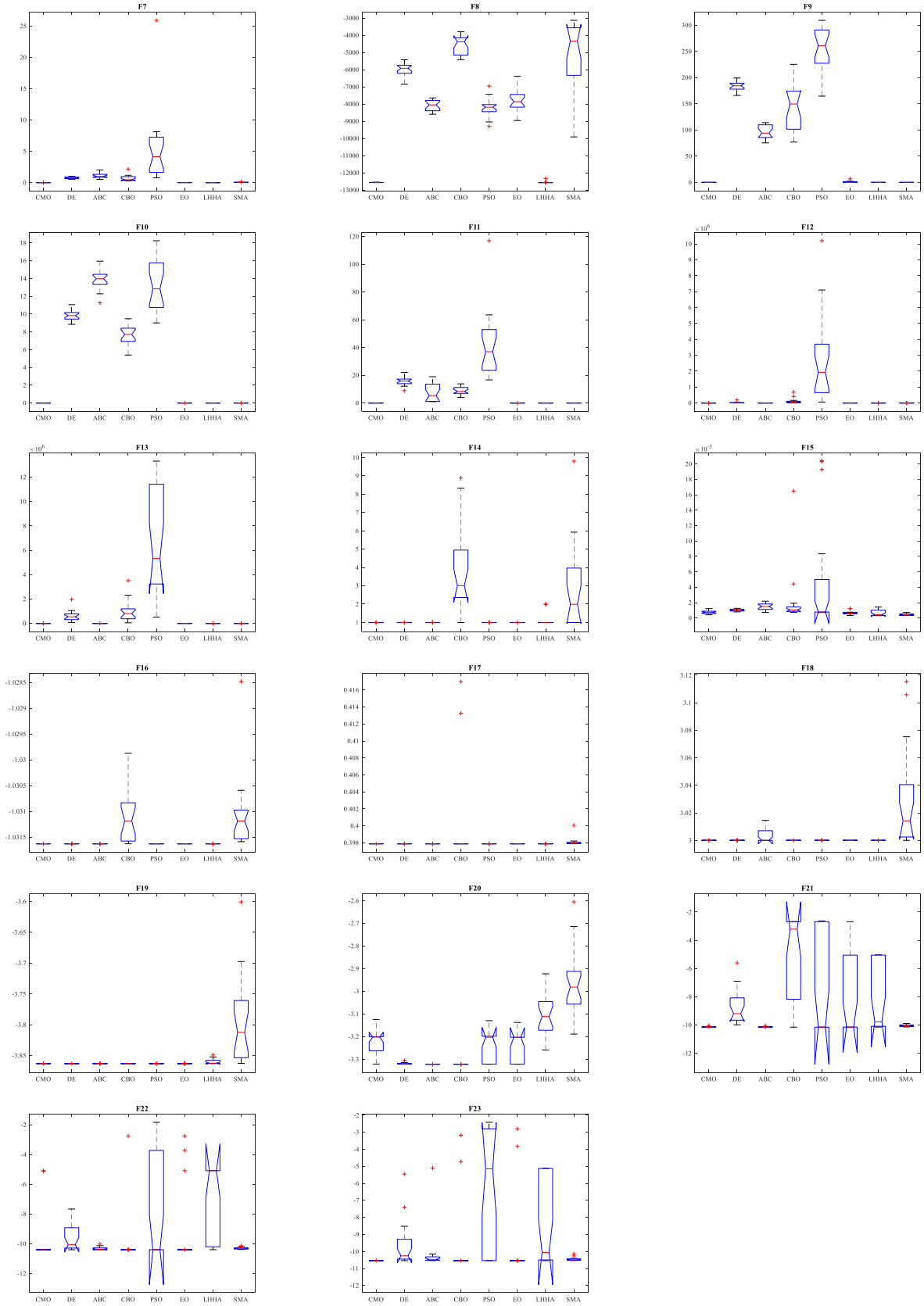


Fig. 6. ANOVA test results for mathematical benchmark functions.

3.3. Engineering truss problems

Trusses are widely used in engineering because of their design flexibility and fast construction [33]. Finding methods to reduce the weight and cost of trusses while adhering to the standard rules and regulations is highly practical. In this section, the performance of the proposed CMO algorithm is evaluated using two benchmark examples of two- and three-dimensional truss optimization problems. The objective function of truss optimization is to minimize the structural weight. Design constraints typically include limitations on element stress and nodal displacements. The optimization problem formulation for truss structures can be expressed as follows:

$$Minimize: W = \sum_{i=1}^{n} \gamma_i l_i X_i, \quad i = 1,2,\dots,n \tag{9}$$

Subject to:

$$\mathrm{g_j^d} = \frac{\mathrm{d_j}}{\bar{\mathrm{d}}_\mathrm{j}} - 1, \quad \mathrm{j} = 1,2,\dots,\mathrm{m}$$

$$g_k^s = \frac{\sigma_k}{\bar{\sigma}_k} - 1, \quad j = 1,2,\dots,n \tag{10}$$

$$X_i^L \le X_i \le X_i^U$$

In the equations above, g_k^s and g_j^d represent the stress and displacement constraints, respectively. W represents the structural weight, X_i, γ_i, and l_i are the cross-sectional area, density of the material, and length of the ith element, respectively. σ_k and d_j represent the stress and displacement of the kth element and jth node, respectively; $\bar{\sigma}_k$ and $\bar{d}_j$ are the allowable values for stress and displacement, respectively. m and n denote the numbers of nodes and elements, respectively.

To handle the constraints in the optimization problem, a multistage assignment penalty function was utilized as an exterior penalty function (EPF). The penalty function is defined as follows:

$$F(x) = f(x) + h(k)H(x), x \in S \subset R^n \tag{11}$$

where k represents the current iteration number, $h(k)$ is the dynamically modified penalty value, $H(x)$ is the penalty factor, and $f(\mathrm{x})$ is the original objective function. $H(x)$ is defined as:

$$H(x) = \sum_{i=1}^{m} \theta\,(q_i(x))\, q_i(x)^{\,\gamma\,(q_i(x))} \tag{12}$$

In equation (12), $q_i(x)$ takes the values of either 0 or $g_i(x)$, where $g_i(x)$ represents the constraint functions and $\theta(q_i(x)$ is a multi-segment assignment function. The penalty function power is denoted as $\gamma(q_i(x))$.

The penalty value is initially set to the lowest possible value, but increases with each iteration as the algorithm progresses [34]. The selection of the penalty coefficient and initial value is crucial in penalty-function methods. If the penalty value is extremely small, the algorithm may generate solutions outside the feasible region. However, if the penalty value is too large, reaching the

boundary of the feasible region becomes challenging and the boundary may not be properly explored. It is common for at least one active constraint to exist at the optimum point, emphasizing the need to search for the entire feasible region, including its boundaries [35].

In this study, following the suggestions in [36], the penalty parameters are set as follows: if $q_i(x) < 1$, then $\gamma\left(q_i(x)\right) = 1$; otherwise, $\gamma\left(q_i(x)\right) = 2$. Additionally, if $q_i(x) < 0.001$, then $\theta\left(q_i(x)\right) = 10$. For the range $0.001 < q_i(x) < 0.1$, $\theta\left(q_i(x)\right) = 20$. For $0.1 < q_i(x) < 1$, $\theta\left(q_i(x)\right) = 100$. otherwise, $\theta\left(q_i(x)\right) = 300$. $h(k)$ is set to $k\sqrt{k}$, where k is the current iteration number.

3.3.1. 10-bar truss example

The 10-bar truss problem is one of the most extensively studied truss problems. The geometrical data for the problem are presented in Fig. 7. The Young's modulus of the material is E = 10^4 ksi and the specific weight is 0.1 lb/in^3. The cross-sectional area of each bar varies between 0.1 and 35 °in^2 while the stress in each bar did not exceed ± 25 ksi. The displacement in each free node in each direction should not exceed 2 inches. As shown in Fig. 7, the vertical load in nodes 2 and 4 is 10^5 lb.

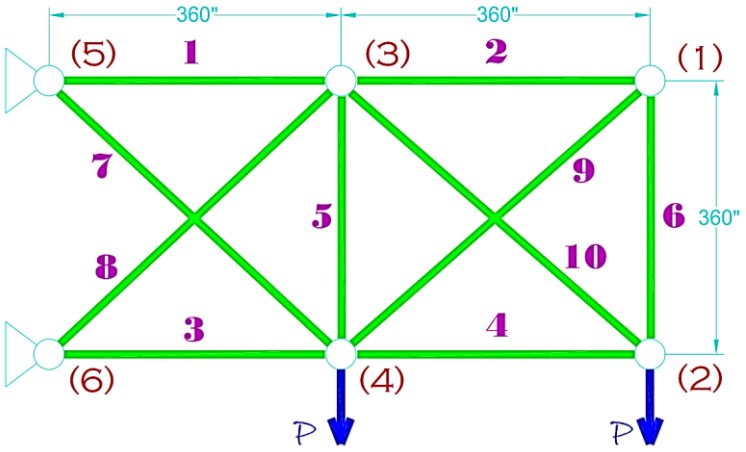


Fig. 7. Schematic presentation of 10-bar truss problem.

During the optimization procedure, a total of 50 particles took part, while the maximum iteration limit was defined at 1000. The optimization outcomes achieved through the CMO approach were juxtaposed with the results of alternative algorithms in Table 9. These findings validate that the CMO algorithm stands out with the lowest essential count of function evaluations (NFEs) and exhibits a narrow standard deviation, ultimately leading to the most optimal solution.

Table 9 presents a comparison of the CMO optimization results with those of other available studies by addressing the number of function evaluations (NFE), optimal design variables, standard deviation, and best and average weights. It is worth mentioning that the maximum displacements, which were assessed using HPSACO in the optimum designs, exceeded the 2.0-inch displacement limits. The suggested CMO converged to a completely feasible design. The CMO algorithm provides an optimized design that can compete with the feasible designs obtained from the ABC-AP and HPSSO.

Table 9
Results of optimization for the 10-bar truss problem.

Design Variables	**WEO** [37]	**HPSSO** [38]	**HPSACO** [39]	**ABC-AP** [40]	**Present Study**
A_1	30.5755	30.5838	30.493	30.548	30.57
A_2	0.1	0.1	0.1	0.1	0.1
A_3	23.3368	23.15103	23.23	23.18	23.08
A_4	15.1497	15.20566	15.346	15.218	15.22
A_5	0.1	0.1	0.1	0.1	0.1
A_6	0.5276	0.548897	0.538	0.551	0.56
A_7	20.9892	21.06437	20.99	21.058	21.04
A_8	7.4458	7.465322	7.451	7.463	7.47
A_9	0.1	0.1	0.1	0.1	0.1
A_{10}	21.5236	21.52935	21.458	21.501	21.56
Best	5060.99	5060.86	5056.56	5060.88	5060.9
Average	5062.09	5062.28	5057.66	N/A	5061.24
Std.	2.05	4.325	1.42	N/A	0.55
NFE	19,540	14,118	10,650	500,000	12,500

3.3.2. 25-bar spatial truss problem

The second problem addressed in this section involves a 25-bar spatial truss structure, depicted in Figure 8. The material density and elasticity modulus for this structure are set at 0.1 lb/in³ and 10,000 ksi, respectively. Each node's maximum allowable displacement in all three coordinate directions is limited to 0.35 inches, while the permissible stress for the members ranges from -40 ksi to +40 ksi. The structural components are categorized into eight groups, denoted as follows: A_1, A_2-A_5, A_6-A_9, A_{10}-A_{11}, A_{12}-A_{13}, A_{14}-A_{17}, A_{18}-A_{21}, and A_{22}-A_{25}.

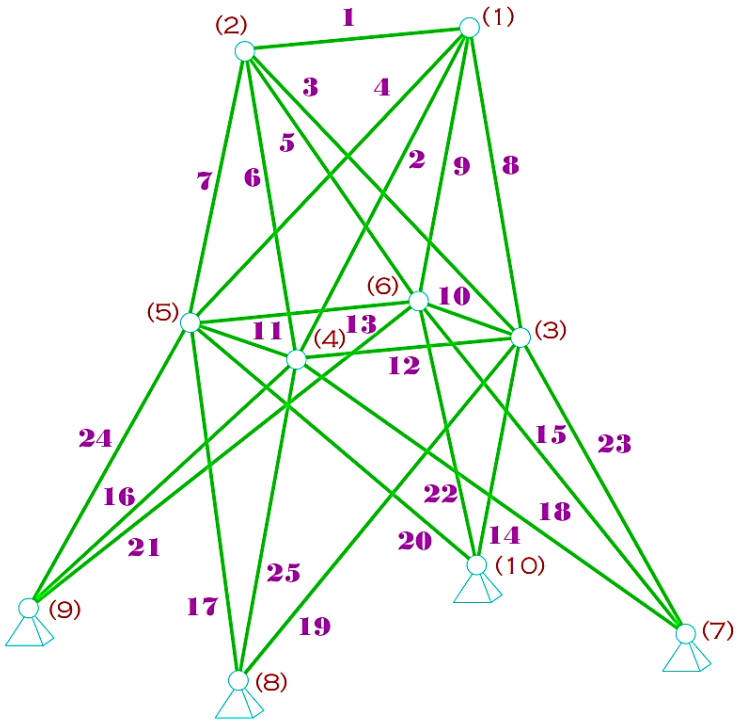


Fig. 8. Schematic presentation of 25-bar spatial truss problem.

The tension and compression stress constraints for each group are presented in Table 10. The cross-sectional areas of each group of elements are assumed variable and change continually from 0.01 in^2 to 3.4 in^2. There are 110 inequality constraints for this problem. The truss is subjected to the loading conditions listed in Table 11, where P_x, P_y, and P_z are the loads along $x-$, $y-$, and $z-$axes, respectively.

Table 10 compares the optimization results of the CMO algorithm, which used 50 particles and 1000 iterations, with those of the BB-BC, EBA, TLBO, and WEO for the 25-bar spatial truss problem described earlier. The results show that all algorithms were able to converge to the best or near-optimal design, but CMO required significantly fewer function evaluations and less computational effort than the other algorithms.

Table 10
Results of optimization for the 25-bar truss.

Element Groups	**BB–BC** [41]	**EBA** [42]	**TLBO** [43]	**WEO** [44]	**Present Study**
A_1	0.01	0.01	0.01	0.01	0.01
$A_2 - A_5$	2.092	1.9789	2.0712	1.9814	1.977
$A_6 - A_9$	2.964	3.0047	2.957	3.0023	3.006
$A_{10} - A_{11}$	0.01	0.01	0.01	0.01	0.01
$A_{12} - A_{13}$	0.01	0.01	0.01	0.01	0.01
$A_{14} - A_{17}$	0.689	0.6888	0.6891	0.6827	0.684
$A_{18} - A_{21}$	1.601	1.6783	1.6209	1.6778	1.680
$A_{22} - A_{25}$	2.686	2.6527	2.6768	2.6612	2.658
Weight (lb)	545.38	545.1688	545.09	545.166	545.18
Average (lb)	545.78	546.4464	545.41	N/A	545.36
SD (lb)	0.491	N/A	0.42	N/A	0.23
NFE	20,566	20,000	15,318	19,750	7,500

Table 11
Loading conditions acting on the 25-bar truss.

Load case	**Nodes**	**Loads**		
		P_x **(kips)**	P_y **(kips)**	P_z **(kips)**
1	1	0.0	20.0	-5.0
	2	0.0	-20.0	-5.0
2	1	1.0	10.0	-5.0
	2	0.0	10.0	-5.0
	3	5.0	0.0	0.0
	6	5.0	0.0	0.0

4. Conclusions

The paper introduces the Center of Mass Optimization (CMO) algorithm, which is a physics-based optimization method inspired by the center of mass of a system of particles. The CMO algorithm incorporates a random walk operator, a mutation operator, and elitism selection to improve local search capabilities and prevent becoming trapped in local optima. One of the

notable features of the CMO algorithm is its lack of adjustable parameters, which makes it easy to implement and use in various engineering applications. The performance of the CMO algorithm was evaluated on 23 mathematical benchmark functions, including unimodal, multimodal, and low-dimensional functions, as well as two truss optimization problems. The experimental results demonstrate that the CMO algorithm outperforms other popular optimization algorithms in terms of efficiency and achieves optimal solutions. Future work can focus on extending the CMO algorithm to handle binary and multi-objective optimization problems, which would further enhance its applicability and versatility. Overall, the CMO algorithm shows promising potential as an effective optimization technique, and its simplicity and strong performance make it a valuable tool for solving engineering optimization problems.

Funding

This research received no external funding.

Conflicts of interest

The authors declare no conflict of interest.

Authors contribution statement

HV: Conceptualization; Data curation; formal analysis; Investigation; Methodology; Software; Validation; Visualization; Roles/Writing – original draft; writing – review, and editing.

References

[1] Varaee H, Shishegaran A, Ghasemi MR. The life-cycle cost analysis based on probabilistic optimization using a novel algorithm. J Build Eng 2021;43:103032. https://doi.org/10.1016/j.jobe.2021.103032.

[2] Sharbatdar MK, Vaez SRH, Amiri GG, Naderpour H. Seismic Response of Base-Isolated Structures with LRB and FPS under near Fault Ground Motions. Procedia Eng 2011;14:3245–51. https://doi.org/10.1016/j.proeng.2011.07.410.

[3] Ahmadi-Nedushan B, Varaee H. Optimal Design of Reinforced Concrete Retaining Walls using a Swarm Intelligence Technique. In: B.H.V. Topping and Y. Tsompanakis, editor. Proc. First Int. Conf. Soft Comput. Technol. Civil, Struct. Environ. Eng., vol. 92, Stirlingshire, Scotland: Civil-Comp Press; 2009, p. 1–12. https://doi.org/10.4203/ccp.92.26.

[4] Ahmadi-Nedushan B, Varaee H. Minimum cost design of concrete slabs using particle swarm optimization with time varying acceleration coefficients. World Appl Sci J 2011;13:2484–94.

[5] Ghasemi MR, Ghiasi R, Varaee H. Probability-based damage detection using kriging surrogates and enhanced ideal gas molecular movement algorithm. ICAVMS 2017 19th Int Conf Acoust Vib Mech Struct 2017;4.

[6] Ghasemi MR, Ghiasi R, Varaee H. Probability-Based Damage Detection of Structures Using Surrogate Model and Enhanced Ideal Gas Molecular Movement Algorithm. Adv Struct Multidiscip Optim 2018;11:1657–74. https://doi.org/10.1007/978-3-319-67988-4_124.

[7] Ghasemi MR, Ghiasi R, Varaee H. Probability-Based Damage Detection of Structures Using Surrogate Model and Enhanced Ideal Gas Molecular Movement Algorithm. Proc. 12th World Congr. Struct. Multidiscip. Optim., Braunschweig, Germany: 2017.

[8] Shabakhty N, Enferadi MH, Ghasemi MR, Varaee H. Application of Shape Memory Alloy Tuned Mass Damper in Vibration Control of Jacket type Offshore Structures. Iran J Mar Sci Technol 2020;7:64–75.

[9] Safari M, Varaee H. Opposition-based ideal gas molecular movement algorithm with Cauchy mutation, velocity clamping, and mirror operator. Expert Syst 2023. https://doi.org/10.1111/exsy.13306.

[10] Ghasemi MR, Varaee H. Modified Ideal Gas Molecular Movement Algorithm Based on Quantum Behavior. Adv. Struct. Multidiscip. Optim., Cham: Springer International Publishing; 2018, p. 1997–2010. https://doi.org/10.1007/978-3-319-67988-4_148.

[11] Ghasemi MR, Varaee H. Enhanced IGMM optimization algorithm based on vibration for numerical and engineering problems. Eng Comput 2018;34:91–116. https://doi.org/10.1007/s00366-017-0523-0.

[12] Ghasemi MR, Varaee H. A fast multi-objective optimization using an efficient ideal gas molecular movement algorithm. Eng Comput 2017;33:477–96. https://doi.org/10.1007/s00366-016-0485-7.

[13] Shishegaran A, Karami B, Safari Danalou E, Varaee H, Rabczuk T. Computational predictions for predicting the performance of steel 1 panel shear wall under explosive loads. Eng Comput (Swansea, Wales) 2021;ahead-of-p. https://doi.org/10.1108/EC-09-2020-0492.

[14] Shishegaran A, Ghasemi MR, Varaee H. Performance of a novel bent-up bars system not interacting with concrete. Front Struct Civ Eng 2019;13:1301–15. https://doi.org/10.1007/s11709-019-0552-4.

[15] Mokarram V, Banan MR. An improved multi-objective optimization approach for performance-based design of structures using nonlinear time-history analyses. Appl Soft Comput J 2018;73:647–65. https://doi.org/10.1016/j.asoc.2018.08.048.

[16] Ghasemi MR, Hinton E, Bulman S. Performance of Genetic Algorithms for Optimization of Frame Structures. In: Parmee I, editor. Adapt. Comput. Des. Manuf. SE - 22, Springer London; 1998, p. 287–99. https://doi.org/10.1007/978-1-4471-1589-2_22.

[17] Kumar P, Pandey S, Maiti PR. A Modified Genetic Algorithm in C++ for Optimization of Steel Truss Structures. J Soft Comput Civ Eng 2021;5:95–108. https://doi.org/10.22115/scce.2021.242552.1249.

[18] Salajegheh E, Heidari A. Optimum design of structures against earthquake by wavelet neural network and filter banks. Earthq Eng Struct Dyn 2005;34:67–82. https://doi.org/10.1002/eqe.417.

[19] Gharehchopogh FS, Nadimi-Shahraki MH, Barshandeh S, Abdollahzadeh B, Zamani H. CQFFA: A Chaotic Quasi-oppositional Farmland Fertility Algorithm for Solving Engineering Optimization Problems. J Bionic Eng 2022. https://doi.org/10.1007/s42235-022-00255-4.

[20] Nadimi-Shahraki MH, Taghian S, Zamani H, Mirjalili S, Elaziz MA. MMKE: Multi-trial vector-based monkey king evolution algorithm and its applications for engineering optimization problems. vol. 18. 2023. https://doi.org/10.1371/journal.pone.0280006.

[21] Akbari M, Henteh M. Comparison of Genetic Algorithm (GA) and Particle Swarm Optimization Algorithm (PSO) for Discrete and Continuous Size Optimization of 2D Truss Structures. J Soft Comput Civ Eng 2019;3:76–97. https://doi.org/10.22115/SCCE.2019.195713.1117.

[22] Varaee H, Ghasemi MR. Engineering optimization based on ideal gas molecular movement algorithm. Eng Comput 2017;33:71–93. https://doi.org/10.1007/s00366-016-0457-y.

[23] Varaee H, Safaeian Hamzehkolaei N, Safari M. A hybrid generalized reduced gradient-based particle swarm optimizer for constrained engineering optimization problems. J Soft Comput Civ Eng 2021;5:86–119. https://doi.org/10.22115/SCCE.2021.282360.1304.

[24] Hoseini Z, Varaee H, Rafieizonooz M, Jay Kim J-H. A New Enhanced Hybrid Grey Wolf Optimizer (GWO) Combined with Elephant Herding Optimization (EHO) Algorithm for Engineering Optimization. J Soft Comput Civ Eng 2022;6:1–42. https://doi.org/10.22115/scce.2022.342360.1436.

[25] Varaee H, Ghasemi MR. An improved chaotic ideal gas molecular movement algorithm for engineering optimization problems. Expert Syst 2022;39. https://doi.org/10.1111/exsy.12913.

[26] Das S, Nagaratnam Suganthan P, Member S. Differential Evolution : A Survey of the State - of - the - Art. Ieee Trans Evol Comput 2011;15:4–31. https://doi.org/10.1109/TEVC.2010.2059031.

[27] Karaboga D, Gorkemli B, Ozturk C. A comprehensive survey : artificial bee colony (ABC) algorithm and applications 2012. https://doi.org/10.1007/s10462-012-9328-0.

[28] Kaveh a., Mahdavi VRR. Colliding Bodies Optimization method for optimum design of truss structures with continuous variables. Adv Eng Softw 2014;70:1–12. https://doi.org/10.1016/j.advengsoft.2014.01.002.

[29] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., Ieee; 1995, p. 39–43. https://doi.org/10.1109/mhs.1995.494215.

[30] Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: A novel optimization algorithm. Knowledge-Based Syst 2020;191:105190. https://doi.org/10.1016/j.knosys.2019.105190.

[31] Kumar V, Sharma V, Naresh R. Leader Harris Hawks algorithm based optimal controller for automatic generation control in PV-hydro-wind integrated power network. Electr Power Syst Res 2023;214:108924.

[32] Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: A new method for stochastic optimization. Futur Gener Comput Syst 2020;111:300–23.

[33] Rezaee Manesh M, Ghasemi SH, Rezaee Manesh M. Dual Target Optimization of Two-Dimensional Truss Using Cost Efficiency and Structural Reliability Sufficiency. J Soft Comput Civ Eng 2020;4:98–111.

[34] Kulkarni O, Kulkarni N, Kulkarni AJ, Kakandikar G. Constrained cohort intelligence using static and dynamic penalty function approach for mechanical components design. Int J Parallel, Emergent Distrib Syst 2018;33:570–88. https://doi.org/10.1080/17445760.2016.1242728.

[35] Grandgirard J, Poinsot D, Krespi L, Nénon JP, Cortesero AM, Miettinen K, et al. Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms. J Glob Optim 2003;27:427–46. https://doi.org/10.1023/A.

[36] Parsopoulos KE, Vrahatis MN, others. Particle swarm optimization method for constrained optimization problems. Intell Technol Appl New Trends Intell Technol 2002;76:214–20.

[37] Kaveh A, Bakhshpoori T. Water Evaporation Optimization: A novel physically inspired optimization algorithm. Comput Struct 2016;167:69–85. https://doi.org/10.1016/j.compstruc.2016.01.008.

[38] Kaveh A, Bakhshpoori T, Afshari E. An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct 2014;143:40–59.

[39] Kaveh a., Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 2009;87:267–83. https://doi.org/10.1016/j.compstruc.2009.01.003.

[40] Sonmez M. Artificial Bee Colony algorithm for optimization of truss structures. Appl Soft Comput J 2011;11:2406–18. https://doi.org/10.1016/j.asoc.2010.09.003.

[41] Özbaşaran H. A study on size optimization of trusses with BB-BC algorithm: review and numerical experiments. Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilim Derg 2018;18:256–64.

[42] Kaveh A, Hosseini SM. Improved bat algorithm based on doppler effect for optimal design of special truss structures. J Comput Civ Eng 2022;36:4022028. https://doi.org/doi.org/10.1061/(ASCE)CP.1943-5487.000104.

[43] Camp CV, Farshchin M. Design of space trusses using modified teaching–learning based optimization. Eng Struct 2014;62–63:87–97. https://doi.org/10.1016/j.engstruct.2014.01.020.

[44] Kaveh A, Bakhshpoori T. An accelerated water evaporation optimization formulation for discrete optimization of skeletal structures. Comput Struct 2016;177:218–28. https://doi.org/10.1016/j.compstruc.2016.08.006.